

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-153037

(43)Date of publication of application : 11.06.1996

(51)Int.Cl.

G06F 12/02

(21)Application number : 06-317678

(71)Applicant : NEC CORP

(22)Date of filing : 28.11.1994

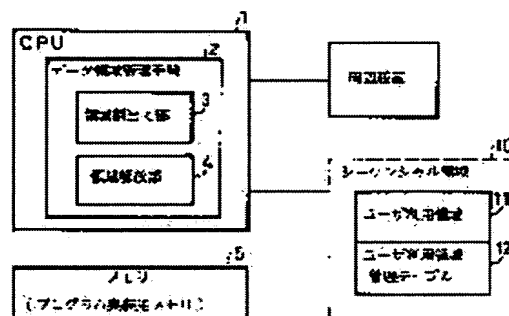
(72)Inventor : KODAMA TOMOAKI

(54) DATA MANAGEMENT SYSTEM IN SEQUENTIAL AREA

(57)Abstract:

PURPOSE: To speed up the allocation/release of a data area in a sequential area to improve the efficiency.

CONSTITUTION: A user use area 11 being the set of data areas used for respective user programs and a user use area management table 12 holding information (use state) showing how the user use area 11 is, is used by the user program are provided in the sequential area 10. The data area management means 12 divides the user use area 11 into the plural blocks of plural kinds of sizes and manages the use state of a block unit in the user use area 11 by the user use area management table 12.



LEGAL STATUS

[Date of request for examination] 28.11.1994

[Date of sending the examiner's decision of rejection] 14.01.1997

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(10)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平8-153037

(43)公開日 平成8年(1996)6月11日

(51)Int.Cl.

G06F 12/02

発明記号

540

庁内整理番号

7623-5B

F I

技術表示箇所

審査請求 有 請求項の数 6 F D (全 10 頁)

(21)出願番号 特願平8-317678
(22)出願日 平成6年(1994)11月28日

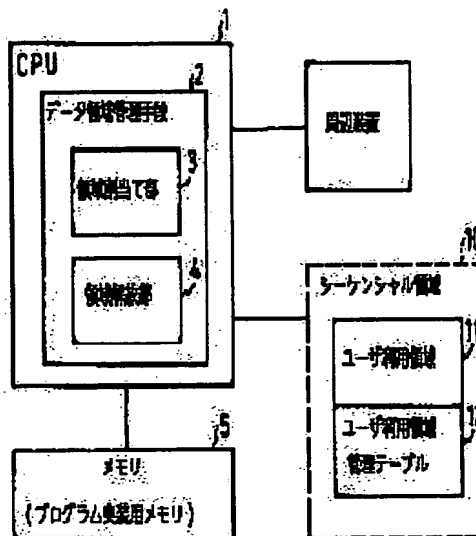
(71)出願人 000004237
日本電気株式会社
東京都港区芝五丁目7番1号
(72)発明者 小玉 知幸
東京都港区芝五丁目7番1号 日本電気株式会社内
(74)代理人 弁護士 河原 純一

(54)【発明の名称】 シーケンシャル領域におけるデータ管理方式

(6)【要約】

【目的】 シーケンシャル領域におけるデータ領域の割当て/解放の高速化および効率化を図る。

【構成】 シーケンシャル領域10内に、各ユーザプログラムに使用されるデータ領域の集合であるユーザ利用領域11と、ユーザ利用領域11がどのようにユーザプログラムに使用されているかを示す情報(使用状態)を保持するユーザ利用領域管理テーブル12とが設けられる。データ領域管理手段2は、ユーザ利用領域11を複数の種類のサイズの複数のブロックに分割し、ユーザ利用領域11におけるブロック単位の使用状態をユーザ利用領域管理テーブル12によって管理する。



【特許請求の範囲】

【請求項1】 シーケンシャル領域内の任意長のデータ領域を使用するユーザプログラムが実行されるコンピュータシステムにおいて、

各ユーザプログラムに使用されるデータ領域の集合であるシーケンシャル領域内のユーザ利用領域と、

このユーザ利用領域がどのようにユーザプログラムに使用されているかを示す情報を保持するシーケンシャル領域内のユーザ利用領域管理テーブルと、

前記ユーザ利用領域を複数の種類のサイズの複数のブロックに分割し、前記ユーザ利用領域におけるブロック単位の使用状態を前記ユーザ利用領域管理テーブルによって管理するデータ領域管理手段とを有することを特徴とするシーケンシャル領域におけるデータ管理方式。

【請求項2】 ユーザプログラムからの領域割当て要求に応じてユーザ利用領域内のブロックを当該ユーザプログラムのために割り当てその割当てを反映してユーザ利用領域管理テーブル内の当該ブロックに対応するエントリに「使用中」を示す情報を設定する領域割当て部と、ユーザプログラムからの領域解放要求に応じてユーザ利用領域管理テーブル内の当該領域解放要求で指定されたブロックに対応するエントリに「未使用」を示す情報を設定する領域解放部とからなるデータ領域管理手段を有することを特徴とする請求項1記載のシーケンシャル領域におけるデータ管理方式。

【請求項3】 割り当てべきブロックのサイズとして、「領域割当て要求に係る要求サイズ以上のブロックのサイズの中で未使用のブロックが存在する最小のサイズ」を採用する領域割当て部を有することを特徴とする請求項2記載のシーケンシャル領域におけるデータ管理方式。

【請求項4】 ブロックのサイズ毎に各サイズ用管理テーブルを備え、「使用状態」および「先頭位置」の項目からなるエントリを有するユーザ利用領域管理テーブルを有することを特徴とする請求項1、請求項2または請求項3記載のシーケンシャル領域におけるデータ管理方式。

【請求項5】 動作するユーザプログラムが固定的なコンピュータシステムに適用されることを特徴とする請求項1、請求項2、請求項3または請求項4記載のシーケンシャル領域におけるデータ管理方式。

【請求項6】 装置制御用コンピュータに係るコンピュータシステムに適用されることを特徴とする請求項5記載のシーケンシャル領域におけるデータ管理方式。

【発明の詳細な説明】

【00001】

【産業上の利用分野】 本発明は、メモリ内に実装されたユーザプログラム（メモリまたはファイル内のシーケンシャルなデータ領域（以下、シーケンシャル領域という）内の任意長（可変長）のデータ領域を使用するユー

ザプログラム）がCPU（Central Processing Unit、中央処理装置）によって実行されるコンピュータシステムにおいて、当該ユーザプログラムのために任意長のデータ領域を当該シーケンシャル領域に動的に割り当てる場合のデータ管理を行うシーケンシャル領域におけるデータ管理方式に関する。

【00002】

【従来の技術】 従来、この種のシーケンシャル領域におけるデータ管理方式では、ユーザプログラムからの領域割当て要求の対象のデータ領域をシーケンシャル領域内の未使用のデータ領域（領域割当て要求の対象の領域サイズに適合する可変長のデータ領域）に確保し、サイズが異なった複数のデータ領域をポインタ等によりチェーン構造（リンク構造）にして管理して、当該可変長のデータ領域群の管理を実現していた。なお、このような方式は、汎用コンピュータに係るコンピュータシステムにおけるOS（Operating System）、すなわちそれぞれが登録商標であるUNIX、MS-DOSまたはOS-9等の汎用OSによって実現される方式である。

【00003】 このような従来のシーケンシャル領域におけるデータ管理方式の管理概念を以下に示す。

【00004】 図5（a）および（b）は、従来のシーケンシャル領域におけるデータ管理方式によるデータ領域の管理概念を説明するための図であり、データ領域の管理におけるデータ領域のチェーン構造を示す図である。

【00005】 図5（a）は、ユーザプログラムがデータ領域に関する領域割当て要求を行う前の状態、すなわちシーケンシャル領域内のデータ領域が確保されている状態を表している。この状態では、使用領域であるデータ領域の各々は、各領域制御ブロックCB～CB内のポインタによってリンクされている。

【00006】 図5（b）は、図5（a）に示す状態において、ユーザプログラムからの領域割当て要求を受け付けた後の状態を示す図である。

【00007】 シーケンシャル領域の管理を行うデータ領域管理部（メモリ内に実装されておりCPUによって実行される処理手順）は、ユーザプログラムから領域割当て要求（所望のデータ領域のサイズの指定を含む要求）を受けると、すでに使用中のデータ領域のチェーン構造を領域制御ブロックに基づいて最後まで追跡し、空き領域（未使用のデータ領域）を探し、当該空き領域を当該ユーザプログラムのために確保する。また、その後、確保したデータ領域を使用中のデータ領域のチェーン構造に追加する。

【00008】 すなわち、図5（b）の場合には、先頭領域（データ領域）に対応する領域制御ブロック（領域制御ブロックCB）からシーケンシャル領域におけるリンク状態を追跡する。つまり、図5（b）に示すように、領域制御ブロックCB～CBを順次に追跡し、

それ以降のデータ領域は未使用であることを認識し、データ領域の後ろに当該領域割当て要求で要求されたサイズのデータ領域を確保し、領域制御ブロック00にデータ領域へのポインタを格納することで、データ領域を使用中のデータ領域のリンク構造に追加する処理を行う。

【0009】また、ユーザプログラムが割当て済みのデータ領域の解放を求める領域解放要求を行った場合には、データ領域管理部は、該当するデータ領域をチェーン構造から排除してそのデータ領域をフリー状態とする処理を行う。

【0010】

【発明が解決しようとする課題】 上述した従来のシーケンシャル領域におけるデータ管理方式では、シーケンシャル領域内の使用中のデータ領域を領域制御ブロックのチェーン構造にて管理しているため、データ領域の割当ておよび解放の際にそのチェーン構造を追跡する必要があった。ここで、データ領域の割当て／解放は一般的に頻繁に行われるため、割当て／解放が繰り返されるとデータ領域を管理するためのチェーン構造が複雑になり、処理の複雑化が生じ、追跡に要する時間が増大するという問題点があった。

【0011】また、いろいろなサイズのデータ領域の割当て／解放が繰り返されると、データ領域のチェーン状態が散乱的になり、空き領域（未使用のデータ領域）が存在するにもかかわらず連続したデータ領域の割当てが不可能な状態が発生するという問題点があった。

【0012】ここで、チェーン構造を構成するデータ領域が散乱的となることによる問題（虫食い状態による問題）に対処するための従来技術として、散乱状態にある未使用のデータ領域をまとめて連続した大きなデータ領域を確保するためのガーベージコレクションを行う技術が存在する。しかし、この技術における処理（ガーベージコレクション処理）では、シーケンシャル領域における大量のデータの移動が必要となるため、大きな処理時間とCPUの処理能力を必要とする。また、ガーベージコレクションを行っている最中には、ユーザプログラムからのデータ領域へのアクセスが不可能となることから、ガーベージコレクション処理中にはユーザプログラムがほぼ停止状態となり、総合的にユーザプログラムの処理性能を著しく低下させるという問題点があった。

【0013】本発明の目的は、上述の点に鑑み、シーケンシャル領域におけるデータ領域の割当て／解放の高速化および効率化を図るシーケンシャル領域におけるデータ管理方式を提供することにある。なお、この発明は、装置制御用コンピュータに係るコンピュータシステム等の「動作するユーザプログラムが固定的なコンピュータシステム」において、特に有効となる。

【0014】なお、データ領域の管理の効率化に関する従来技術としては、特開昭63-141147号公報

（可変長ブロック領域のメモリ管理制御方式）が公表されている。しかし、この従来技術は、「ブロック管理情報領域が利用者の使用するブロック領域とは分離され別領域で管理されている」ことや、「ブロックの使用状態管理テーブルと可変長ブロックを管理するための管理テーブルを持ち、ビットで管理すること」により、可変長ブロック管理処理が非常に簡単になり、処理の高性能化が図れる」という効果を達成するものであり（当該公報中の第329頁左下欄【発明の効果】参照）、取り扱われるデータ領域のブロックのサイズはあくまでも固定的である。したがって、当該従来技術は、複数の種類のサイズの複数のブロックによってシーケンシャル領域のデータ管理を行う本願発明とは、その構成等を異にしている。

【0015】

【課題を解決するための手段】 本発明のシーケンシャル領域におけるデータ管理方式は、シーケンシャル領域内の任意長のデータ領域を使用するユーザプログラムが実行されるコンピュータシステムにおいて、各ユーザプログラムに使用されるデータ領域の集合であるシーケンシャル領域内のユーザ利用領域と、このユーザ利用領域がどのようにユーザプログラムに使用されているかを示す情報を保持するシーケンシャル領域内のユーザ利用領域管理テーブルと、前記ユーザ利用領域を複数の種類のサイズの複数のブロックに分割し、前記ユーザ利用領域におけるブロック単位の使用状態を前記ユーザ利用領域管理テーブルによって管理するデータ領域管理手段とを有する。

【0016】

【作用】 本発明のシーケンシャル領域におけるデータ管理方式では、各ユーザプログラムに使用されるデータ領域の集合であるユーザ利用領域がシーケンシャル領域内に設けられ、ユーザ利用領域がどのようにユーザプログラムに使用されているかを示す情報（使用状態）を保持するユーザ利用領域管理テーブルがシーケンシャル領域内に設けられ、データ領域管理手段がユーザ利用領域を複数の種類のサイズの複数のブロックに分割しユーザ利用領域におけるブロック単位の使用状態をユーザ利用領域管理テーブルによって管理する。

【0017】

【実施例】 次に、本発明について図面を参照して詳細に説明する。

【0018】 図1は、本発明のシーケンシャル領域におけるデータ管理方式の実施例の構成を示すブロック図である。

【0019】 本実施例のシーケンシャル領域におけるデータ管理方式は、CPU1と、ユーザプログラムおよびデータ領域管理手段2（シーケンシャル領域10を管理するための処理手順を実現するプログラム）を実装するプログラム実装用メモリであるメモリ5と、シーケンシ

ャル領域10とを含んで構成されている。なお、メモリ5内のプログラム（データ領域管理手段2を実現するプログラム等）がCPU1によって実行されることによって当該プログラムの処理手順が実行されるが、このブロック図においては、便宜上、CPU1内にデータ領域管理手段2を示している。また、本実施例のシーケンシャル領域におけるデータ管理方式が適用されるコンピュータシステムには、周辺装置も存在し得る（ただし、以下の説明においては周辺装置については言及しない）。

【0020】CPU1は、領域割当て部3および領域解放部4を備えるデータ領域管理手段2を含んで構成されている。

【0021】図2は、シーケンシャル領域10の構成を示す図であり、ユーザ利用領域11およびユーザ利用領域管理テーブル12の構成と両者の関係とを示す図である。

【0022】シーケンシャル領域10は、ユーザプログラムによって利用される（ユーザプログラムがデータの書き込み/読出しを実際に行う）データ領域群であるユーザ利用領域11と、ユーザ利用領域11をブロック化してその使用状態を管理するためのユーザ利用領域管理テーブル12との2つの領域に分けられる。データ領域管理手段2は、ユーザ利用領域管理テーブル12を使用して、ユーザプログラムのために割り当てられるデータ領域（ブロック）の管理を行う。

【0023】ユーザ利用領域11は複数の種類のサイズ（サイズA、サイズB、...、サイズN、...）の複数のブロックに分割され、それぞれのサイズに対応する各管理テーブル（サイズA用管理テーブル、サイズB用管理テーブル、...、サイズN用管理テーブル、...）がユーザ利用領域管理テーブル12内に作成される。ユーザ利用領域管理テーブル12内の各エントリは、ユーザ利用領域11の各ブロックの先頭アドレスを示す「先頭位置」とそのブロックが使用中（"use"）であるか未使用（"nouse"）であるかを識別するための情報である「使用状態」とを保持している。

【0024】ユーザ利用領域11の分割方法（サイズの種類の数および各サイズに対応するブロック数等の決定方法）は、各ユーザプログラムが必要とするデータ領域の大きさに基づいて、あらかじめ設定される。したがって、本実施例のシーケンシャル領域におけるデータ管理方式が適用されるコンピュータシステムは、動作するユーザプログラムが固定的であることが望ましい。

【0025】図3は、領域割当て時のデータ領域管理手段2の処理（領域割当て部3の処理）を示す流れ図である。この処理は、ブロックサイズ決定ステップ301と、ユーザ利用領域管理テーブル検索ステップ302と、目的サイズ空き領域存在判定ステップ303と、目的サイズ超過空き領域存在判定ステップ304と、「領域なし」応答ステップ305と、ブロック割当てステップ306と、アドレス応答ステップ307とからなる。

【0026】図4は、領域解放時のデータ領域管理手段2の処理（領域解放部4の処理）を示す流れ図である。

この処理は、指定アドレス確認ステップ401と、アドレス正当性判定ステップ402と、エラー応答ステップ403と、ブロック使用状態確認ステップ404と、使用状態判定ステップ405と、エラー応答ステップ406と、使用状態変更ステップ407と、正常応答ステップ408とからなる。

【0027】次に、このように構成された本実施例のシーケンシャル領域におけるデータ管理方式の動作について説明する。

【0028】第1に、図3を参照して、領域割当て時の動作（領域割当て部3の処理手順）を説明する。

【0029】メモリ5内に実装されたユーザプログラム（ユーザ利用領域11内のデータ領域を使用するユーザプログラム）がCPU1によって実行され、シーケンシャル領域10内のユーザ利用領域11内のデータ領域の割当てを求める要求（領域割当て要求）が当該ユーザプログラムによって行われると、データ領域管理手段2内の領域割当て部3は以下のような処理を行ってユーザ利用領域11内のデータ領域を当該ユーザプログラムのために確保する（図3参照）。

【0030】まず、ユーザプログラムからの領域割当て要求に係る要求サイズ（当該ユーザプログラムが必要とするデータ領域のサイズ）を基に、割り当てるべきユーザ利用領域11内のブロックのサイズ（目的サイズ）を決定する（ステップ301）。この決定方法としては、要求サイズ以上のブロックのサイズの中で最小のサイズを目的サイズとして決定する方法が採られる。

【0031】その後、ユーザ利用領域管理テーブル12を検索し（ステップ302）、ステップ301で決定した目的サイズを有するブロックの中で空き領域のもの（未使用ブロック）が存在するかどうかを判定する（ステップ303）。

【0032】ステップ303で「未使用ブロックが存在しない（目的サイズのブロックが全て使用中である）」と判定した場合には、1ランク大きな種類のサイズのブロックの中から未使用ブロックを検索する。この検索を1ランク毎に順に大きくしていき、目的サイズより大きなブロックの全てを対象として繰り返すことにより、ユーザ利用領域11に目的サイズより大きなサイズのブロックが存在するかどうかを判定する（ステップ304）。

【0033】ステップ304で「目的サイズより大きなサイズのブロックの全てが使用中である」と判定した場合には、当該ユーザプログラムに対して「領域なし」

（メモリ不足で目的サイズのデータ領域の割当てが不可能である旨）を応答し（ステップ305）、処理を終了する。

【0034】ステップ303またはステップ304で「目的サイズ以上の未使用ブロックが存在する」と判定した場合には、ユーザ利用領域11内の当該未使用ブロックを当該ユーザプログラムのために割り当て（確保し）、当該未使用ブロックに対応するユーザ利用領域管理テーブル12内のエントリ中の「使用状態」を使用中（“use”）に設定し（ステップ306）、当該未使用ブロックのアドレス（先頭アドレス）をユーザプログラムに対して応答し（ステップ307）、処理を終了する。

【0035】このように、領域割当て時にデータ領域のブロックのサイズをデータ領域管理手段2が自動的に決定するので、ユーザプログラムはシーケンシャル領域10がブロック化されていることを意識する必要がない。ここで、仮にユーザプログラムの要求サイズと一致するサイズのブロックが全て使用中であっても、要求サイズより大きなサイズのブロックを当該ユーザプログラムに自動的に割り当てることが可能となるので、シーケンシャル領域10の使用効率は低下するが、当該ユーザプログラムが動作しないという事態を回避することはできる。

【0036】第2に、図4を参照して、領域解放時の動作（領域解放部4の処理手順）を説明する。

【0037】データ領域の解放を求める要求（領域解放要求）がユーザプログラムによって行われると、データ領域管理手段2内の領域解放部4は以下のような処理を行って領域解放要求の対象のユーザ利用領域11内のブロック（データ領域）を解放する（図4参照）。

【0038】まず、ユーザプログラムからの領域解放要求において指定されたアドレス（解放対象のブロックを指定するための情報）を確認し（ステップ401）、指定されたアドレスの正当性（当該アドレスがユーザ利用領域11内のアドレスの範囲に該当するか否か等）をチェックし、当該アドレスが不正でないかどうかを判定する（ステップ402）。

【0039】ステップ402で「当該アドレスが不正である」と判定した場合には、当該ユーザプログラムにエラー応答（指定アドレスが不正である旨のエラー応答）を返し（ステップ403）、処理を終了する。

【0040】ステップ402で「当該アドレスが正当である」と判定した場合には、そのアドレスによって特定されるブロックに対応するユーザ利用領域管理テーブル12内のエントリ（そのアドレスを先頭アドレスとして「先頭位置」の項目に有するエントリ）における「使用状態」を確認し（ステップ404）、「使用状態」が“use”（使用中）であるか“no-use”（未使用）であるかを判定する（ステップ405）。

【0041】ステップ405で「使用状態」が「未使用」であると判定した場合には、当該ユーザプログラムにエラー応答（指定ブロックが未使用である旨のエラ

ー応答）を返し（ステップ406）、処理を終了する。

【0042】ステップ405で「使用状態」が「使用中」であると判定した場合には、ユーザ利用領域管理テーブル12内の当該エントリ中の「使用状態」を「未使用」（“no-use”）に変更し（ステップ407）、当該ユーザプログラムに正常応答を返し（ステップ408）、処理を終了する。

【0043】

【発明の効果】汎用的に使用されるOS（汎用コンピュータにおけるOS）の場合には、OS上で動作するユーザプログラム（OS以外のプログラム）が不特定なため、ユーザプログラムがOSに要求するデータ領域のサイズも特定することができず、あらゆるサイズに対応できるようにする必要がある。したがって、チェーン構造にてシーケンシャル領域内のデータ領域を管理する方式（図5参照）は、ある意味では有効な管理方式といえる。

【0044】しかしながら、前述のように、このようなチェーン構造にてシーケンシャル領域内のデータ領域を管理する方式では、データ領域の割当て／解放が繰り返されると、データ領域のチェーン構造が随機的になるため、ガーベージコレクションの必要が生じ、ユーザプログラムの処理性能の低下を招く（ガーベージコレクション処理によるユーザプログラムの停止状態が生じる）。

【0045】これに対して本発明のシーケンシャル領域におけるデータ管理方式では、データ領域を可変長のブロック単位で管理することにより、データ領域の離散状態（虫食い状態）が発生しなくなり、ガーベージコレクションの必要がなくなつてユーザプログラムの処理性能が向上するという効果がある。

【0046】また、チェーン構造にてシーケンシャル領域内のデータ領域を管理する方式では、データ領域の割当て／解放を行う場合にチェーン構造におけるポインタの追跡が必要となり、データ領域のチェーン構造が複雑になればなるほど、処理の複雑化を招き、データ領域の割当て／解放時の処理時間が長くなる。さらに、チェーン構造にてシーケンシャル領域内のデータ領域を管理する方式では、万一ユーザプログラムが誤って領域制御ブロックを変更してしまうと、チェーン構造が破壊されてシーケンシャル領域内の全てのデータ領域が使用不可能となってしまう。

【0047】これに対して、本発明のシーケンシャル領域におけるデータ管理方式では、データ領域の管理のためにチェーン構造を利用していないので、ユーザ利用領域におけるブロックの設定（ブロック分割の態様）が固定的になり、ユーザ利用領域管理テーブルを操作するのみでシーケンシャル領域の管理が可能となり、以下の～に示すような効果が生じる。

【0048】ユーザ利用領域内のブロック（データ領域）の割当ての際には、ユーザ利用領域管理テーブル

を探索することだけで割り当てるべきブロックを決定できるため、従来のようにチェーン構造のポインタの追跡等を必要とせず、データ領域の割当ての際の処理性能を向上させること（処理の簡易化および処理時間の短縮化）ができる。

【0049】 ブロック（データ領域）の解放時に、従来のように「使用中のデータ領域のチェーン構造からの当該ブロックを排除する操作」の必要がなくなり、ユーザ利用領域管理テーブル内の情報だけを変更すればよいことから、データ領域の解放の際の処理性能を向上させること（処理の簡易化および処理時間の短縮化）が可能となる。

【0050】 ユーザプログラムが使用するデータ領域群（ユーザ利用領域）とデータ領域管理手段が管理する領域（ユーザ利用領域管理テーブル）とが分離しているので、万一ユーザプログラムが自己のために割り当てられたデータ領域以外を操作したとしても、管理情報が破壊されるおそれが少ないため、そのような場合における被害を最小に抑えることが可能となる。

【0051】 ところで、装置制御用コンピュータに係るコンピュータシステム等の「動作するユーザプログラムが固定的なコンピュータシステム」においては、汎用コンピュータに係るコンピュータシステムと比べ、ユーザプログラムが動作するために必要とするデータ領域のブロックのサイズの種類およびブロック数はほぼ固定的になる。一方、本発明のシーケンシャル領域におけるデータ管理方式は、あらかじめユーザプログラムが必要とするブロックのサイズの種類およびブロック数を設定してユーザ利用領域のブロック分割の態様を決めて、シーケ

ンシャル領域内の各データ領域の有効利用を図っている。したがって、本発明は、「動作するユーザプログラムが固定的なコンピュータシステム」に適用されることによって、特に、データ領域の割当て／解放の高速化および効率化を図ることができるという効果が顕著となる。

【図面の簡単な説明】

【図1】 本発明のシーケンシャル領域におけるデータ管理方式の一実施例の構成を示すブロック図である。

【図2】 図1中のシーケンシャル領域の構成を示す図である。

【図3】 図1に示すシーケンシャル領域におけるデータ管理方式における領域割当て部の処理を示す流れ図である。

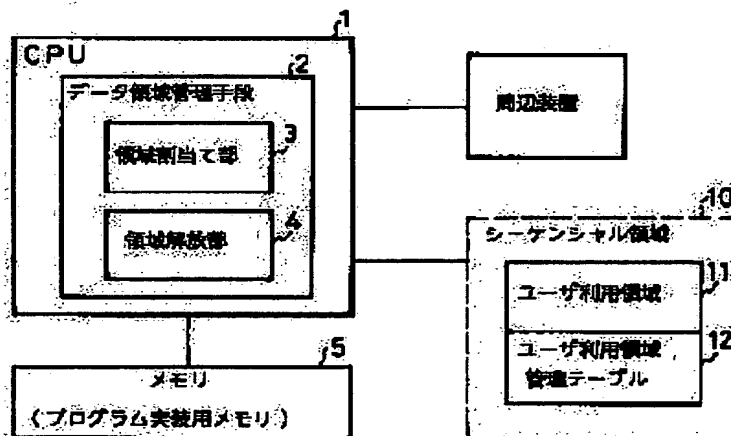
【図4】 図1に示すシーケンシャル領域におけるデータ管理方式における領域解放部の処理を示す流れ図である。

【図5】 従来のシーケンシャル領域におけるデータ管理方式によるデータ領域の管理概念を説明するための図である。

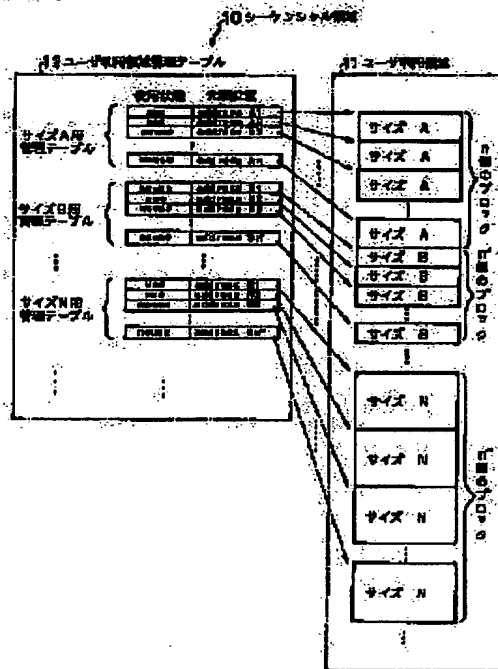
【符号の説明】

- 1 CPU
- 2 データ領域管理手段
- 3 領域割当て部
- 4 領域解放部
- 5 メモリ（プログラム実装用メモリ）
- 10 シーケンシャル領域
- 11 ユーザ利用領域
- 12 ユーザ利用領域管理テーブル

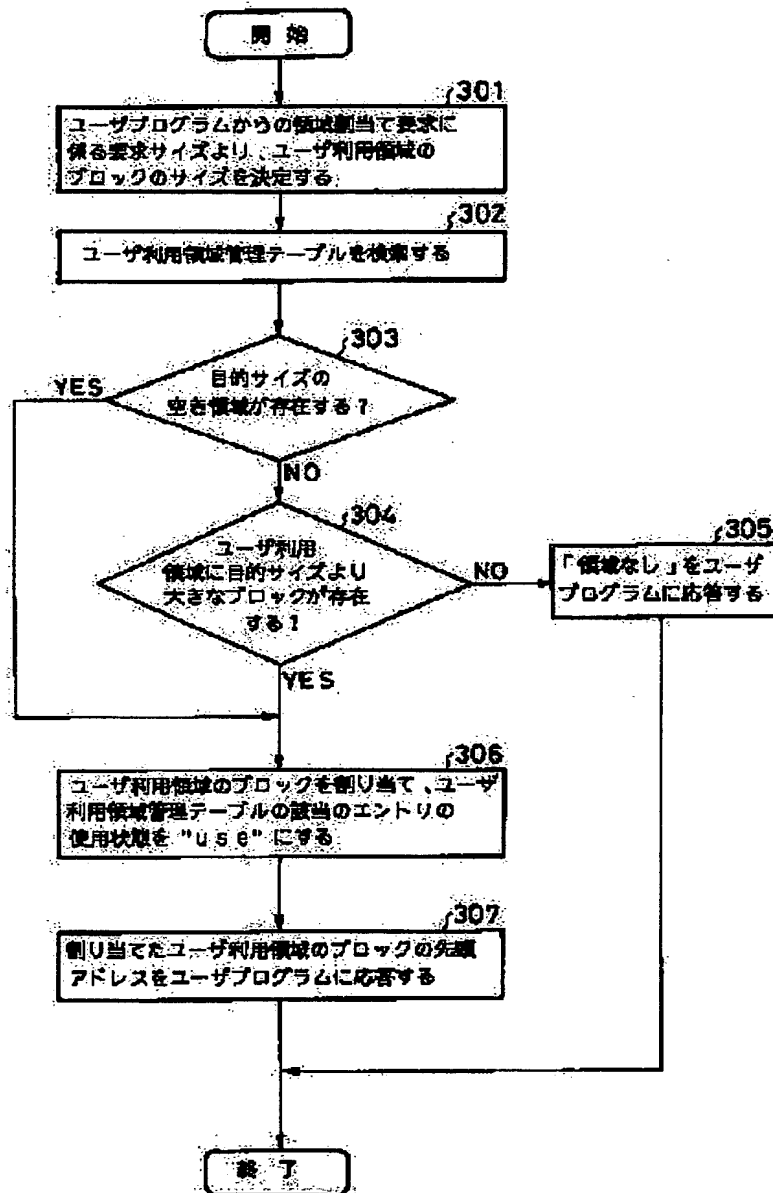
【図1】



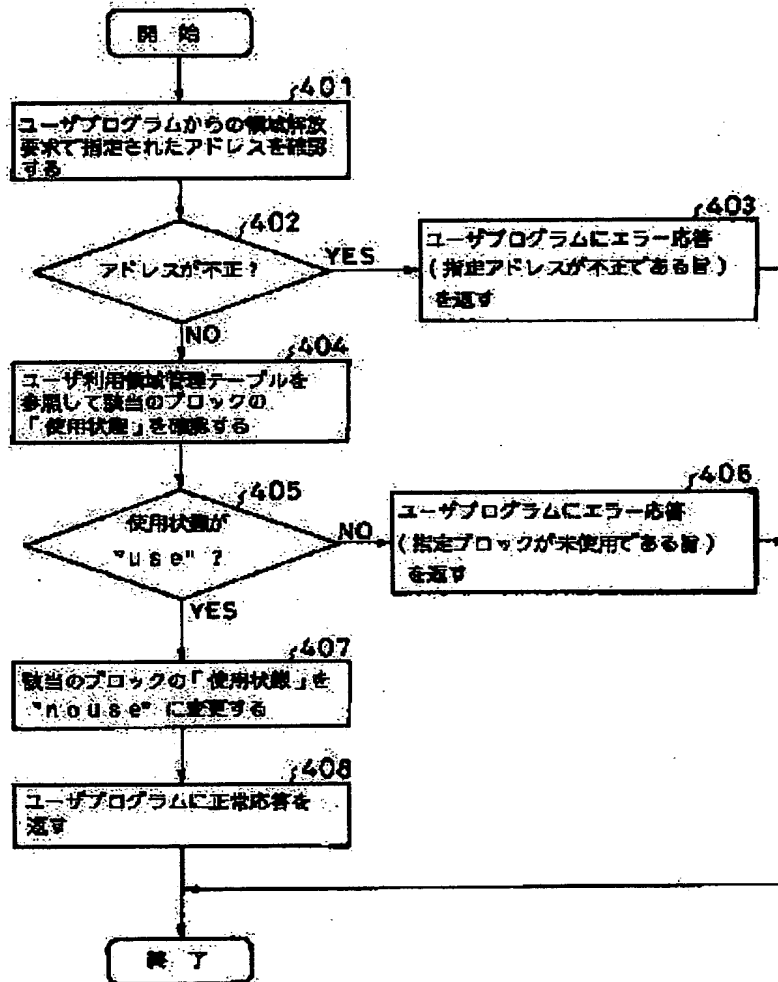
【図2】



[図.9]

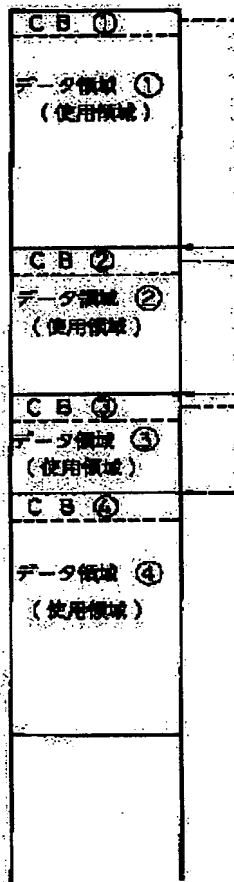


【図4】



【図5】

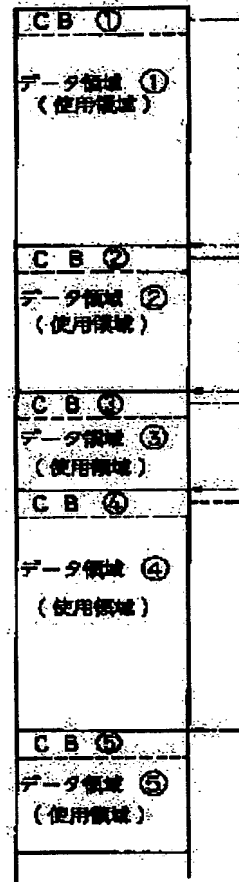
(a)



! CB:領域制御ブロック

データ領域 ④ までチェーンに組み込んだ状態

(b)



! CB:領域制御ブロック

データ領域 ⑤ をチェーンに組み込んだ後の状態